Adversarial Separation and Adaptation Network for Far-Field Speaker Verification

Lu Yi, Man Wai Mak

Dept. of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China
lu-louisa.yi@connect.polyu.hk, enmwmak@polyu.edu.hk

Abstract

Typically, speaker verification systems are highly optimized on the speech collected by close-talking microphones. However, these systems will perform poorly when the users use far-field microphones during verification. In this paper, we propose an adversarial separation and adaptation network (ADSAN) to extract speaker discriminative and domain-invariant features through adversarial learning. The idea is based on the notion that speaker embedding comprises domain-specific components and domain-shared components, and that the two components can be disentangled by the interplay of the separation network and the adaptation network in the ADSAN. We also propose to incorporate a mutual information neural estimator into the domain adaptation network to retain speaker discriminative information. Experiments on the VOiCES Challenge 2019 demonstrate that the proposed approaches can produce more domain-invariant and speaker discriminative representations, which could help to reduce the domain shift caused by different types of microphones and reverberant environments.

Index Terms: Far field speaker verification; domain adaptation; adversarial learning; domain mismatch

1. Introduction

Today’s speaker recognition systems have achieved remarkable performance under controlled environments such as quiet offices and clean telephone channels. However, far-field speaker recognition is still challenging because the environmental noise and reverberation effect are hard to control. Due to the difference in microphone characteristics, there is a domain mismatch between near-field microphone speech and far-field microphone speech. The mismatch can make a speaker recognition system that is trained on near-field microphone speech to perform poorly on far-field microphone speech. Domain adaptation (DA) can be applied to address this problem.

Domain adaptation algorithms for speaker verification (SV) either adapt the probabilistic linear discriminant analysis (PLDA) models to fit the target data [1] or find a common embedding space in which the feature distributions have low discrepancy across multiple domains. For the former, a recent approach is to use the concept of correlation alignment [2] to estimate a pseudo-in-domain covariance matrix. The matrix is then interpolated with the out-of-domain covariance matrix of the PLDA model [3]. Finding a common embedding space is a more general approach in that it is independent of the backend. Research in this direction has focused on compensating for the inter-data set variability [4], projecting out the inter-domain features and clean telephone channels. However, far-field speaker recognition is still challenging because the environmental noise and reverberation effect are hard to control. Due to the difference in microphone characteristics, there is a domain mismatch between near-field microphone speech and far-field microphone speech. The mismatch can make a speaker recognition system that is trained on near-field microphone speech to perform poorly on far-field microphone speech. Domain adaptation (DA) can be applied to address this problem.

Domain adaptation algorithms for speaker verification (SV) either adapt the probabilistic linear discriminant analysis (PLDA) models to fit the target data [1] or find a common embedding space in which the feature distributions have low discrepancy across multiple domains. For the former, a recent approach is to use the concept of correlation alignment [2] to estimate a pseudo-in-domain covariance matrix. The matrix is then interpolated with the out-of-domain covariance matrix of the PLDA model [3]. Finding a common embedding space is a more general approach in that it is independent of the backend. Research in this direction has focused on compensating for the inter-data set variability [4], projecting out the inter-domain features and clean telephone channels. However, far-field speaker recognition is still challenging because the environmental noise and reverberation effect are hard to control. Due to the difference in microphone characteristics, there is a domain mismatch between near-field microphone speech and far-field microphone speech. The mismatch can make a speaker recognition system that is trained on near-field microphone speech to perform poorly on far-field microphone speech. Domain adaptation (DA) can be applied to address this problem.

Domain adaptation algorithms for speaker verification (SV) either adapt the probabilistic linear discriminant analysis (PLDA) models to fit the target data [1] or find a common embedding space in which the feature distributions have low discrepancy across multiple domains. For the former, a recent approach is to use the concept of correlation alignment [2] to estimate a pseudo-in-domain covariance matrix. The matrix is then interpolated with the out-of-domain covariance matrix of the PLDA model [3]. Finding a common embedding space is a more general approach in that it is independent of the backend. Research in this direction has focused on compensating for the inter-data set variability [4], projecting out the inter-domain features and clean telephone channels. However, far-field speaker recognition is still challenging because the environmental noise and reverberation effect are hard to control. Due to the difference in microphone characteristics, there is a domain mismatch between near-field microphone speech and far-field microphone speech. The mismatch can make a speaker recognition system that is trained on near-field microphone speech to perform poorly on far-field microphone speech. Domain adaptation (DA) can be applied to address this problem.

Domain adaptation algorithms for speaker verification (SV) either adapt the probabilistic linear discriminant analysis (PLDA) models to fit the target data [1] or find a common embedding space in which the feature distributions have low discrepancy across multiple domains. For the former, a recent approach is to use the concept of correlation alignment [2] to estimate a pseudo-in-domain covariance matrix. The matrix is then interpolated with the out-of-domain covariance matrix of the PLDA model [3]. Finding a common embedding space is a more general approach in that it is independent of the backend. Research in this direction has focused on compensating for the inter-data set variability [4], projecting out the inter-domain features and clean telephone channels. However, far-field speaker recognition is still challenging because the environmental noise and reverberation effect are hard to control. Due to the difference in microphone characteristics, there is a domain mismatch between near-field microphone speech and far-field microphone speech. The mismatch can make a speaker recognition system that is trained on near-field microphone speech to perform poorly on far-field microphone speech. Domain adaptation (DA) can be applied to address this problem.

Domain adaptation algorithms for speaker verification (SV) either adapt the probabilistic linear discriminant analysis (PLDA) models to fit the target data [1] or find a common embedding space in which the feature distributions have low discrepancy across multiple domains. For the former, a recent approach is to use the concept of correlation alignment [2] to estimate a pseudo-in-domain covariance matrix. The matrix is then interpolated with the out-of-domain covariance matrix of the PLDA model [3]. Finding a common embedding space is a more general approach in that it is independent of the backend. Research in this direction has focused on compensating for the inter-data set variability [4], projecting out the inter-domain features and clean telephone channels. However, far-field speaker recognition is still challenging because the environmental noise and reverberation effect are hard to control. Due to the difference in microphone characteristics, there is a domain mismatch between near-field microphone speech and far-field microphone speech. The mismatch can make a speaker recognition system that is trained on near-field microphone speech to perform poorly on far-field microphone speech. Domain adaptation (DA) can be applied to address this problem.

Domain adaptation algorithms for speaker verification (SV) either adapt the probabilistic linear discriminant analysis (PLDA) models to fit the target data [1] or find a common embedding space in which the feature distributions have low discrepancy across multiple domains. For the former, a recent approach is to use the concept of correlation alignment [2] to estimate a pseudo-in-domain covariance matrix. The matrix is then interpolated with the out-of-domain covariance matrix of the PLDA model [3]. Finding a common embedding space is a more general approach in that it is independent of the backend. Research in this direction has focused on compensating for the inter-data set variability [4], projecting out the inter-domain features and clean telephone channels. However, far-field speaker recognition is still challenging because the environmental noise and reverberation effect are hard to control. Due to the difference in microphone characteristics, there is a domain mismatch between near-field microphone speech and far-field microphone speech. The mismatch can make a speaker recognition system that is trained on near-field microphone speech to perform poorly on far-field microphone speech. Domain adaptation (DA) can be applied to address this problem.

Domain adaptation algorithms for speaker verification (SV) either adapt the probabilistic linear discriminant analysis (PLDA) models to fit the target data [1] or find a common embedding space in which the feature distributions have low discrepancy across multiple domains. For the former, a recent approach is to use the concept of correlation alignment [2] to estimate a pseudo-in-domain covariance matrix. The matrix is then interpolated with the out-of-domain covariance matrix of the PLDA model [3]. Finding a common embedding space is a more general approach in that it is independent of the backend. Research in this direction has focused on compensating for the inter-data set variability [4], projecting out the inter-domain features and clean telephone channels. However, far-field speaker recognition is still challenging because the environmental noise and reverberation effect are hard to control. Due to the difference in microphone characteristics, there is a domain mismatch between near-field microphone speech and far-field microphone speech. The mismatch can make a speaker recognition system that is trained on near-field microphone speech to perform poorly on far-field microphone speech. Domain adaptation (DA) can be applied to address this problem.

Domain adaptation algorithms for speaker verification (SV) either adapt the probabilistic linear discriminant analysis (PLDA) models to fit the target data [1] or find a common embedding space in which the feature distributions have low discrepancy across multiple domains. For the former, a recent approach is to use the concept of correlation alignment [2] to estimate a pseudo-in-domain covariance matrix. The matrix is then interpolated with the out-of-domain covariance matrix of the PLDA model [3]. Finding a common embedding space is a more general approach in that it is independent of the backend. Research in this direction has focused on compensating for the inter-data set variability [4], projecting out the inter-domain features and clean telephone channels. However, far-field speaker recognition is still challenging because the environmental noise and reverberation effect are hard to control. Due to the difference in microphone characteristics, there is a domain mismatch between near-field microphone speech and far-field microphone speech. The mismatch can make a speaker recognition system that is trained on near-field microphone speech to perform poorly on far-field microphone speech. Domain adaptation (DA) can be applied to address this problem.

Domain adaptation algorithms for speaker verification (SV) either adapt the probabilistic linear discriminant analysis (PLDA) models to fit the target data [1] or find a common embedding space in which the feature distributions have low discrepancy across multiple domains. For the former, a recent approach is to use the concept of correlation alignment [2] to estimate a pseudo-in-domain covariance matrix. The matrix is then interpolated with the out-of-domain covariance matrix of the PLDA model [3]. Finding a common embedding space is a more general approach in that it is independent of the backend. Research in this direction has focused on compensating for the inter-data set variability [4], projecting out the inter-
thogonal to each other, a discriminator is used to maximize the differences between the shared properties and domain-specific properties. It is assumed that the representations are different if they can always be classified correctly by an optimal discriminator.

One potential risk of DSNs is that the features produced by the shared encoder may not contain enough task-related information (in our case, the speaker information), especially when the decoder is flexible enough [15, 16]. Therefore, we further propose to maximize the mutual information between the input and output of the shared encoder to retain as much speaker information as possible in the common features (z^s_1 and z^t_1 in Fig. 1). Using labelled closed-talking utterances and unlabelled far-field utterances, the proposed network can be trained to reduce the domain mismatch caused by different types of microphones. The resulting network can improve the performance of speaker verification systems that are trained on near-field microphone speech but evaluated on far-field microphone speech.

2. Adversarial Separation and Adaptation Network

2.1. Network Structure and Loss Functions

Let X = \{ (x_i, y_{spk}) \}_{i=1}^{N_x} \) represents a labelled dataset of \( N_x \) samples from the source domain \( D_s \) and \( X' = \{ x'_i \}_{i=1}^{N_x} \) represents an unlabelled dataset of \( N_x \) samples from the target domain \( D_t \). Apart from applying the original DSN proposed by Bousmalis et al. [13], we further propose a variant of DSN called adversarial separation and adaptation network (ADSAN), as Fig. 1 shows.

The proposed network comprises three encoders, three discriminators, and a decoder. The shared encoder \( G_h \) is trained adversarially to extract the common features \( \{ z^s_1, z^t_1 \} \) from both source and target domains. The speaker discriminator is applied to make the common features to be speaker discriminative. Because only the source data have speaker labels, the speaker discriminator is trained to minimize the classification loss on the source domain. The domain-dependent encoders, \( G_s \) and \( G_t \), are trained to extract domain-specific features \( \{ z^s_2, z^t_2 \} \), which are assumed to comprise as much domain-specific information as possible. The shared decoder \( R \) is trained to reconstruct the original features \( x' \) or \( x \) from the concatenation of the domain-specific features and shared features \( \{ z^s_2, z^t_2 \} \) or \( \{ z^s_3, z^t_3 \} \). The reconstruction loss back-propagated to the source and target encoders can force the extracted features to maintain the information of the original data. To ensure that the common features can be disentangled from the domain-dependent information, the separation discriminator \( D_{sep} \) is applied to discriminate the vectors \( z^s_3 \) and \( z^t_3 \) into three groups: shared, source, and target. The adaptation discriminator \( D_{adv} \) is utilized to distinguish whether the shared features are from the source domain or from the target domain. Meanwhile, the shared encoder tries to produce shared features that make the adaptation discriminator into believing that there is no difference between \( z^s_1 \) and \( z^t_1 \). As a result, domain-invariant and speaker discriminative features can be extracted from the output of the shared encoder, which are expected to improve the speaker verification performance on the target domain.

To achieve the objectives mentioned above, we need to reduce the classification error on the source domain by minimizing the cross-entropy loss:

\[
\mathcal{L}_{task} = \mathbb{E}_{x \sim D_s} \left[ -\sum_{k=1}^{K} y_{spk}^{(k)} \log p_{spk}^{(k)}(h(x)) \right].
\]

where the subscript \( k \) denotes the \( k \)-th output of the speaker discriminator.

The shared decoder is trained to minimize the mean squared error:

\[
\mathcal{L}_{recon} = \mathbb{E}_{x \sim D_s} \left\| x - R \left( \left[ G_s \left( x \right) \right] \right) \right\|_2^2
+ \mathbb{E}_{x' \sim D_t} \left\| x' - R \left( \left[ G_t \left( x' \right) \right] \right) \right\|_2^2.
\]

The separation discriminator has three outputs, which correspond to \( y_{sep}^{(1)} = P(\text{shared}|x), y_{sep}^{(2)} = P(\text{source}|x), \) and \( y_{sep}^{(3)} = P(\text{target}|x), \) respectively. Therefore, it is trained to minimize the following cross-entropy loss:

\[
\mathcal{L}_{sep} = \mathbb{E}_{x \sim D_s} \left[ -\log D_{sep} \left( G_s \left( x \right) \right) \right] + \mathbb{E}_{x' \sim D_t} \left[ -\log D_{sep} \left( G_t \left( x' \right) \right) \right],
\]

where the subscripts 1, 2, and 3 correspond to shared features, source-domain dependent features, and target-domain dependent features, respectively.

The adaptation discriminator has a sigmoid output to determine whether the shared features come from the source domain or target domain. Its loss function is

\[
\mathcal{L}_{adv} = -\mathbb{E}_{x \sim D_s} \log D_{adv} \left( G_s \left( x \right) \right) - \mathbb{E}_{x' \sim D_t} \log \left[ 1 - D_{adv} \left( G_s \left( x' \right) \right) \right].
\]

The total loss for training the ADSAN is

\[
\mathcal{L}_{ADSAN} = \mathcal{L}_{task} + \alpha \mathcal{L}_{sep} - \beta \mathcal{L}_{adv} + \gamma \mathcal{L}_{recon}.
\]

And the minmax optimization can be summarized as:

\[
\min_{\theta_{spk}, \theta_{sep}, \theta_{s}, \theta_{t}} \max_{\phi_{s}, \phi_{t}, \phi_{sep}, \phi_{adv}} \mathcal{L}_{ADSAN} \left( \theta_{spk}, \theta_{sep}, \theta_{s}, \theta_{t}, \phi_{s}, \phi_{t}, \phi_{sep}, \phi_{adv} \right),
\]

where \( \theta_{spk}, \theta_{sep}, \theta_{s}, \theta_{t}, \phi_{s}, \phi_{t}, \phi_{sep}, \phi_{adv} \) are the parameters of the speaker discriminator, separation discriminator, adaptation discriminator, source encoder, target encoder, shared encoder, and shared decoder, respectively.
The difference between the DSN and our proposed ADSAN is the definition of $\mathcal{L}_{\text{sep}}$. For the DSN, the difference between domain-specific representations and the common representations is determined by orthogonality, i.e.,

$$
\mathcal{L}_{\text{DSN}} = \|z_s' - z_s^*\|^2_p + \|z_t' - z_t^*\|^2_p.
$$

(7)

For ADSAN, on the other hand, the difference is defined by the classification loss in Eq. 3.

### 2.2. Mutual Information Neural Estimator

As mentioned previously, a flexible decoder may prevent the shared features $\{z_s', z_t'\}$ from retaining task-related information. One possible solution is to maximize the mutual information between the encoder’s output and the encoder’s input.

Mutual information is a measure of information shared between random variables. If two variables are independent, the mutual information is zero and high mutual information indicates high dependency between two random variables. However, mutual information is tractable only for discrete random variables or continuous random variables with known probability distributions. Traditional approaches are non-parametric [17, 18, 19, 20] or rely on the approximate Gaussianity of data distributions [21]. These approaches cannot scale well with sample size or dimension [22]. To address this issue, Belgazi et al. [23] proposed a mutual information neural estimator (MINE) to estimate mutual information. They demonstrated that MINE can help to improve the training of adversarial networks. Therefore, we applied the MINE to estimate the mutual information.

MINE utilizes a deep neural network with parameters $\theta \in \Theta$ to find a lower bound of the mutual information:

$$
I(X; Z) \geq I_{\theta}(X, Z),
$$

(8)

where $I_{\theta}(X, Z)$ is defined as

$$
I_{\theta}(X, Z) = \sup_{\theta \in \Theta} \mathbb{E}_{X \sim p_X}[T_\theta] - \log \left( \mathbb{E}_{X \sim p_X} [e^{T_\theta}] \right).
$$

(9)

In Eq. 9, $T_\theta : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ is a function parametrized by the deep neural network. The expectations in Eq. 9 are estimated using empirical samples from $p_{X, Z}$ and $p_X \otimes p_Z$, where $p_{X, Z}$ is the joint distribution and $p_X \otimes p_Z$ is the product of the marginal distributions. Alternatively, it can be done by shuffling the samples from the joint distribution along the batch axis.

The training of MINE is realized by minimizing the loss

$$
\mathcal{L}_{\text{MINE}}(x, z) = -I_{\theta}(x^s; G_h(x^s)).
$$

(10)

By incorporating MINE into the ADSAN, we can calculate the lower bound of the mutual information between $x^s$ and $z^s_s$ as well as $x^t$ and $z^s_t$. Therefore, the shared encoder is trained to maximize the estimated mutual information:

$$
I_{\theta}(x, z) = \sigma_1 I_{\theta}(x^s, G_h(x^s)) + \sigma_2 I_{\theta}(x^t, G_h(x^t)),
$$

(11)

where

$$
I_{\theta}(x^s, G_h(x^s)) = \sup_{\theta \in \Theta} \mathbb{E}_{p(x^s, z_s^*)}[T_\theta(x^s, G_h(x^s))] - \log \mathbb{E}_{p(x^s, z_s^*)} [e^{T_\theta(x^s, G_h(x^s))}]
$$

(12)

$$
I_{\theta}(x^t, G_h(x^t)) = \sup_{\theta \in \Theta} \mathbb{E}_{p(x^t, z_t^*)}[T_\theta(x^t, G_h(x^t))] - \log \mathbb{E}_{p(x^t, z_t^*)} [e^{T_\theta(x^t, G_h(x^t))}]
$$

(13)

And the total loss of the adversarial separation and adaptation network with a mutual information neural estimator is

$$
\mathcal{L}_{\text{ADSAN}} = \mathcal{L}_{\text{task}} + \alpha \mathcal{L}_{\text{sep}} - \beta \mathcal{L}_{\text{adv}} + \gamma \mathcal{L}_{\text{recon}} - I_{\theta}(x, z).
$$

(14)

The parameters of components in the networks are trained with the following minimax optimization:

$$
\min_{\theta_x, \theta_z, \theta_h, \phi_x, \phi_z, \phi_h, \phi_t} \max_{\theta_d} \mathcal{L}_{\text{ADSAN}}(\theta_x, \theta_z, \theta_h, \phi_x, \phi_z, \phi_h, \phi_t, \theta_d).
$$

(15)

### 3. Experimental Setting

X-vectors extracted from near-field microphone speech and far-field microphone speech were respectively used as the source and target domain data to train the ADSAN to produce domain-invariant features. The domain-invariant features from the source domain were used to train a PLDA backend. Then the trained PLDA backend was used to test the domain-invariant features from the target domain.

#### 3.1. Datasets

The source domain comprises utterances from VoxCeleb1 [24] and VoxCeleb2 [25] to conform to the fixed condition of the VOiCES Challenge 2019 [26]. The combination of VoxCeleb1, VoxCeleb2, and their augmented sets were used to train an x-vector extractor. The augmented sets were obtained using MUSAN [27] based on the Kaldi’s receipts. There are ~2.3M utterances spoken by 7,323 speakers from the source domain. The trained x-vector extractor was then used to extract x-vectors of the utterances from the target domain.

The target domain comprises utterances from the VOiCES Challenge 2019, which consists of a development set and an evaluation set. There are 15,904 utterances from 196 speakers in the development set, and 11,392 utterances in the evaluation set.

When training the adaptation networks, the source domain training data were selected from the original VoxCeleb1 & 2 without augmentation. Therefore, the utterances from the source domain can be considered as “clean”, while the utterances from the target domain are “noisy”. The number of utterances per speaker in VoxCeleb1 & 2 ranges from 20 to 500, which may cause imbalance training in the speaker discriminator ($D_{sp}$). To address this issue, at least $M_{\text{min}}$ and at most $M_{\text{max}}$ utterances with the highest signal-to-noise ratios (SNRs) for each speaker from the source domain were selected as source training data. This selection also helps to aggravate the mismatch between the source and target domains, thereby making the results more meaningful. Inspired by [12], we estimated the SNRs using the WADA-SNR algorithm [28]. In our experiments, $M_{\text{max}}$ is the minimum number of utterances of valid speakers, which is 21; $M_{\text{max}}$ is the median number of utterances of valid speakers, which is 124. For the target domain, we used the utterances in the development set to adapt the ADSAN model and tested the performance on the evaluation set.

#### 3.2. Baseline Systems

In addition to comparing with a baseline system that uses raw x-vectors, i.e., without any transformations, we also compared our proposed network with some other networks, including a domain adaptation neural network (DANN) [29], a DANN with MINE, and a domain separation network (DSN) [13]. Referring to Eq. 14 and Eq. 11, if the weight parameters $\alpha$, $\gamma$, $\sigma_1$, and $\sigma_2$ are set to zero, then the proposed network becomes a DANN; if setting $\sigma_1$ and $\sigma_2$ to zero and replacing $\mathcal{L}_{\text{sep}}$ (Eq. 3) with Eq. 7, a DSN can be obtained. These three systems were used as the baseline systems in this paper, and the configurations of the sub-networks in the DANN and the DSN are the same as those in
the proposed network.

3.3. Networks without MINE

To exclude the MINE in the training, \( \sigma_1 \) and \( \sigma_2 \) in Eq. 11 were set to 0. The remaining weight parameters for the ADSAN and the DSN were set as follows: \( \alpha = 1.0, \beta = 0.1, \) and \( \gamma = 1.0; \) for DANN, \( \alpha = \gamma = \sigma_1 = \sigma_2 = 0. \) The sub-networks are fully connected neural networks with two hidden layers, each with 1024 neurons; the separation discriminator is the exception, which has one hidden layer and 100 neurons. The learning rate for all sub-networks is 0.0001 and batch normalization was applied to the three encoders and the shared decoder. The activation function in all sub-networks is the leaky ReLU.

3.4. Networks with MINE

Systems improved upon the DSN and the ADSAN were created by incorporating a MINE into the loss function. The weight parameters in the loss function (Eq. 14 and Eq. 11) were set to \( \alpha = 1.0, \beta = 0.1, \gamma = 1.0, \sigma_1 = 0.2, \) and \( \sigma_2 = 0.4. \) The MINE is a deep neural network whose size depends on the size of data being estimated. In our experiments, the MINE has two hidden layers, each with 100 neurons. Before training the adaptation models, the MINE was pre-trained with the whole training data from the source domain for 5 epochs. Then, for each epoch in the main training loop, the adaptation and separation networks (comprising \( G_h, G_s, G_r, R, D_q, D_a, \) and \( D_{sp} \)) and the MINE were trained consecutively, i.e., when the adaptation and the separation networks were trained, the MINE is frozen, and vice versa when the MINE was trained. Because mutual information is unbounded, it could be infinitely large. We applied gradient clipping to prevent infinite gradient. This strategy can ensure that the networks can be updated normally. The initial learning rate of the MINE is 0.0001, which decays every 1000 steps at a decay rate of 0.96. The learning rate for the other sub-networks is 0.00005.

3.5. PLDA Training and Scoring

As a pre-processing step, we projected the raw x-vector to a 200-dimensional space by linear discriminant analysis followed by length normalization. Then, we used the pre-processed x-vectors derived from Voxceleb1 & 2 (with and without data augmentation) to train the PLDA models of our baseline systems (the first row of Tables 1 and 2). We performed the same pre-processing on the transformed x-vectors obtained by various domain adaptation networks (DANN, DSN, and ADSAN) to train other sets of PLDA models. For scoring, the pre-processed x-vectors and the network-transformed pre-processed x-vectors were derived from VOiCES data and were fed to the respective PLDA models. When computing the PLDA scores, the x-vectors were centered by using the mean x-vector of the enrollment data in the development set of VOiCES.

4. Results and Discussions

For fair comparisons, the same preprocessing was applied to all the systems. In addition, the label information of the target domain data was not used during the entire training process.

Table 1 and Table 2 show the performance on the development set and the evaluation set obtained by different systems, respectively. It can be observed that the domain adaptation networks are more effective for the evaluation set in which the utterances are more noisy and are subject to more severe reverberation. The DANN with MINE achieves the worst performance, which indicates a negative transfer. Compared with the DANN, the DSN and ADSAN can achieve better results, and both the EER and minDCF are further reduced when the MINE was incorporated into the systems. When training the MINE in the DANN, the mutual-information loss always converges to a very small value (closed to 0). This indicates that the MINE cannot play a useful role in the DANN.

For the evaluation set, the DSN with MINE achieves the lowest EER while the ADSAN with MINE achieves the lowest minDCF. Different from the DSN, the ADSAN applies a moderate constraint on the difference between domain-specific components and domain-shared components (Eq. 3 vs. Eq. 7). Investigations via t-SNE plots (not shown) produced by the DSN- and ADSAN-transformed vectors suggest that the distributions of the ADSAN-transformed vectors are closer to the distributions of the original vectors; they also exhibit a smaller distance between the source domain and the target domain. These results reveal that the proposed ADSAN can produce domain-invariant and speaker discriminative representations, which are beneficial for speaker verification.

5. Conclusions

In this paper, we propose an adversarial separation and adaptation network motivated by the domain separation network. We also propose to incorporate a mutual information neural estimator into these two domain separation networks. The proposed approaches can enforce the shared encoder to disentangle the domain-invariant features from the domain-specific properties. Our experiments on the VOiCES corpus show that the proposed approaches can outperform the DANN. And training the networks with the consideration of mutual information can further increase the speaker information in the extracted features.
6. References


